An Improved Formulation for Model Predictive Control of Legged Robots for Gait Planning and Feedback Control

An Improved Formulation for Model Predictive Control of Legged Robots for Gait Planning and Feedback Control

Kai Yuan and Zhibin Li

***Abstract*— Predictive control methods for walking commonly use low dimensional models, such as a Linear Inverted Pendulum Model (LIPM), for simplifying the complex dynamics of legged robots. This paper identifies the physical limitations of the modeling methods that do not account for external disturbances, and then analyzes the issues of numerical stability of Model Predictive Control (MPC) using different models with variable receding horizons. We propose a new modeling formulation that can be used for both gait planning and feedback control in an MPC scheme. The advantages are the improved numerical stability for long prediction horizons and the robustness against various disturbances. Benchmarks were rigorously studied to compare the proposed MPC scheme with the existing ones in terms of numerical stability and disturbance rejection. The effectiveness of the controller is demonstrated in both MATLAB and Gazebo simulations.**

I. Introduction

The performances achieved by legged robots during the DARPA Robotics Challenge (DRC) suggest that more improvements in control shall still be made [1]. The common approach for locomotion among all high ranked DRC humanoids is to split motion planning and stabilizing control into two separate parts [2], [3]. First, an optimal walking pattern under the given environmental constraint is found and used as reference in a feed-forward manner. Then a closed-loop control tracks this reference for the execution, i.e. stabilization of the planned motion pattern under disturbances and uncertainties. This decoupled use of feed-forward and feedback control leads to suboptimal performances, especially in the presence of disturbances.

This motivates the use of Model Predictive Control (MPC) to remedy this problem [4]. MPC achieves closed-loop control by continuously solving the feed-forward optimization using the feedback of current states and constraints. In MPC, high-level tasks can be embedded in the cost function of the optimization problem, and stability of the motion is ensured by designing suitable constraints. This study aims to formulate a coherent model for MPC suitable for both online/offline planning as well as realtime feedback control against external perturbations.

Zero-Moment Point (ZMP) based approaches generate stable gaits by keeping the ZMP within the Support Polygon (SP) [5]. This is usually enforced by either closely tracking the ZMP, or by constraining the ZMP in an optimization problem. When the solver time is secondary for planning, more complete and complex whole-body models could be used to plan for a long horizon resulting in a small deviation from the optimal trajectory. In contrast, MPC solves the optimization problem using simple models, and stability around the planned trajectory is guaranteed by optimizing at a high frequency. The most common simple model is the Linear Inverted Pendulum Model (LIPM) [6]. Despite its limitations on the Center of Mass (COM) height and the necessity of coplanar contacts between feet and ground, it is extensively used in optimization for dynamic locomotion: a Differenial Dynamic Programming (DDP) approach was proposed in [7], and the work in [8] used the LIPM to calculate the closed-form solution of an LQR problem. Lastly, the LIPM dynamics can also be used to represent the ZMP constraint [9] and to design feedback controllers [3], [10].

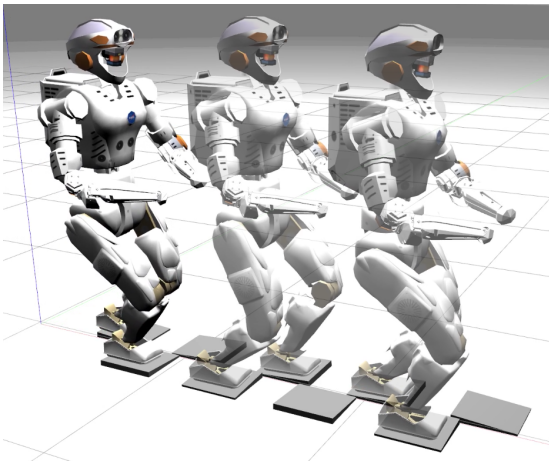


Fig. 1: Physics simulation in Gazebo of NASA's Valkyrie walking over uneven terrain using the proposed MPC framework.

In the past decades, the importance of MPC for dynamic locomotion has increased. First an unconstrained MPC scheme was proposed in [11] for walking pattern generation of bipedal humanoids. Improved disturbance rejection was then achieved by constraining the ZMP [12]. Furthermore MPC was used for automatic foot placement [13], Push Recovery [14], and Capture Point (CP) tracking [15], [16].

We propose an MPC framework that particularly distinguishes the external perturbation in our formulation making it real-world applicable, and also has dual-uses for Motion Planning and Feedback Control. Our formulation is able to generate a stable COM motion given Center of Pressure (COP) or CP references while satisfying the ZMP constraints. Our contributions are summarized as follows:

- Analysis of numerical and physical issues in existing MPC frameworks;

Kai Yuan and Zhibin Li are with the School of Informatics, The University of Edinburgh, UK.
Email: {kai.yuan, zhibin.li}@ed.ac.uk

- A proposed formulation that solves the numerical and physical problems analyzed above;
- Integration of the proposed MPC with whole-body control for real-world application (Fig. 1);
- Benchmarking of the proposed MPC framework against existing ones.

In Section II, the different use of dynamics models for optimization are revisited and compared, and a better formulation is proposed to explicitly differentiate the COM acceleration caused by external pushes. An MPC framework for Walking Pattern Generation and Feedback Control is proposed in Section III. Then the numerical stability of different optimization formulations are analyzed, evaluated, and discussed in Section IV. A comparison of these controllers in the presence of disturbances is then studied in Section V. Lastly we conclude our study in Section VI.

II. Dynamic Models

The ability to model the robot's dynamics allows control design to provide stability and robustness to uncertainties. The idea of the LIPM is based on the assumption that the dominant dynamics of a biped robot can be described by a single inverted pendulum [6] (Fig. 2), which is fast-solvable for optimization problems.

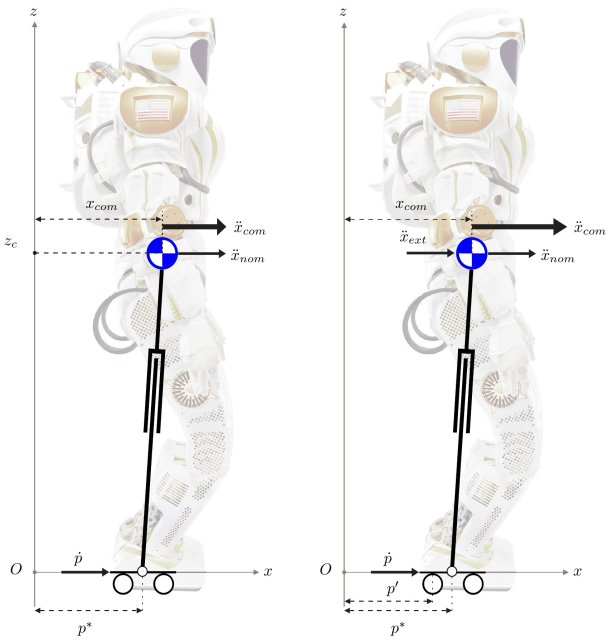


Fig. 2: Representation of NASA's Valkyrie's dynamics as an LIPM. The real COP p^* is controlled by COP velocity $\dot{p}$ through a torque-control based tracking controller. The resultant COM acceleration $\ddot{x}_{com}$ is composed by the acceleration generated by the real COP and the external disturbance $\ddot{x}_{ext}$ (left: no disturbance, right: with disturbance). The fictitious ZMP p' is represented by the CTM.

A. Linear Inverted Pendulum Model: COP as Control Input

For a constant COM height z_c, the dynamics of the Inverted Pendulum become linear and the states of the COM are determined by a second order differential equation [11]:

$$\ddot{x} = \frac{g}{z_c}(x - p_x), \tag{1}$$

with COM position x, velocity $\dot{x}$, acceleration $\ddot{x}$, height z_c, gravity constant g, and ZMP p_x. The LIPM describes the motion of the COM propelled by the Center of Pressure (COP). Due to symmetry of the LIPM motion, only the x component is analysed; the y component behaves analogously.

In the literature the LIP based model (1) was represented with both a two-dimensional $\boldsymbol{x} = [x, \dot{x}]^T$ [15], [17], and a three-dimensional state space vector $\boldsymbol{x} = [x, \dot{x}, p]^T$ [18], [14]. The two-dimensional state space version is used for Capture Point (CP) tracking with the output $y = [1, \sqrt{z_c/g}]\boldsymbol{x}$. The three-dimensional version was used both for control design [18], [14] and state estimation [19]. The ZMP's dynamics exhibit an integrator behavior using the rate change $\dot{p}$ directly as input into the system.

B. Cart-Table Model: Jerk as Control Input

A common way to plan a stable gait is to predefine a ZMP trajectory within the Support Polygon that will be then tracked closely, where the Cart-Table Model (CTM) [11] serves as an inverse dynamics model of the LIPM (1) that enables accurate ZMP tracking by the output feedback of the simulated ZMP p given COM position x and acceleration $\ddot{x}$:

$$p = x - \frac{z_c}{g}\ddot{x}. \tag{2}$$

Although the CTM is suitable for gait planning, it has two limitations: **1.** its input-output causality does not reflect the causality of the actual robot: the real system does not have a real control input that can be arbitrarily manipulated at the COM. In fact, the real system behaves like an Inverted Pendulum, where the COM is driven by the COP. **2.** Due to the reversed causality the COP (ZMP) position is calculated wrongly in the case of disturbance. The real COP p^*, which can be measured and directly controlled, generates the nominal COM state $\boldsymbol{x}_{nom} = [x_{nom}, \dot{x}_{nom}, \ddot{x}_{nom}]$ (Fig. 2 left). For the undisturbed system, COM state $\boldsymbol{x}_{com}$ and the COM motion generated by COP $\boldsymbol{x}_{nom}$ are identical, i.e. $\boldsymbol{x}_{com} = \boldsymbol{x}_{nom}$. However, in case of an external disturbance (Fig. 2 right) that causes acceleration $\ddot{x}_{ext}$, the COM state evolves according to the dynamics generated by the resulted acceleration $\ddot{x}_{com} = g(x_{com} - p^*)/z_c + \ddot{x}_{ext}$, while the COP p^* is controlled by a COP tracking controller and independent from the external push. The CTM uses the resultant COM state to rule back the cause of this acceleration, i.e. where the COP/ZMP is. However, once the external acceleration is injected, the CTM no longer accurately captures this behavior and thus yields a wrong COP calculation as $p' = x_{com} - z_c(\ddot{x}_{GRF} + \ddot{x}_{ext})/g = p^* - z_c\ddot{x}_{ext}/g$.

C. Proposed formulation

The fact that the COM state does not generally reflect the COP of a real system due to disturbance motivates the incorporation of the COP as an additional state in order to correctly model the real physical process. By differentiating

(1), we can see that the *jerk* is determined by COM velocity and rate change of COP $\dot{p}$:

$$\dddot{x} = \frac{g}{z_c}(\dot{x} - \dot{p}). \tag{3}$$

Also, by rearranging (3), the rate change of COP $\dot{p}$ can be represented by the COM state $\dot{x}$ and the COM *jerk* $\dddot{x}$ resulted by the influence of the COP position:

$$\dot{p} = \dot{x} - \frac{z_c}{g}\dddot{x}. \tag{4}$$

In theory, both $\dot{p}$ and $\dddot{x}$ are mutually exchangeable control inputs to represent the newly introduced dynamics (4) of p :

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{g}{z_c} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\frac{g}{z_c} \\ 1 \end{bmatrix}\dot{p} \tag{5}$$

$$y_{COP} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}\boldsymbol{x} \tag{6}$$

$$y_{CP} = \begin{bmatrix} 1 & \sqrt{\frac{z_c}{g}} & 0 & 0 \end{bmatrix}\boldsymbol{x}. \tag{7}$$

Our proposed equivalent formulation is:

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ p \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ -\frac{z_c}{g} \end{bmatrix}\dddot{x} \tag{8}$$

$$y_{COP} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}\boldsymbol{x} \tag{9}$$

$$y_{CP} = \begin{bmatrix} 1 & \sqrt{\frac{z_c}{g}} & 0 & 0 \end{bmatrix}\boldsymbol{x}. \tag{10}$$

To sum up, the COM dynamics can be controlled by two different control efforts: controlling the COP p (1) or its derivative $\dot{p}$, or controlling the jerk $\dddot{x}$ (2). Both models require a COP controller that is able to track the desired COP, which is compatible with modern torque controlled hardware. The different formulations that were used in this study are summarised in Table I.

#	Model	Control Input	Origin
A	LIPM: $\boldsymbol{x} = [x, \dot{x}]^T$	p	[17]
B	LIPM: $\boldsymbol{x} = [x, \dot{x}]^T$	p	[15]
C	LIPM: $\boldsymbol{x} = [x, \dot{x}, p]^T$	$\dot{p}$	[14], [18]
D'	LIPM: $\boldsymbol{x} = [x, \dot{x}, \ddot{x}, p]^T$	$\dot{p}$	Proposed (5)
D	LIPM: $\boldsymbol{x} = [x, \dot{x}, \ddot{x}, p]^T$	$\dddot{x}$ (representing $\dot{p}$)	Proposed (8)
E	CTM: $\boldsymbol{x} = [x, \dot{x}, \ddot{x}]^T$	$\dddot{x}$	[11], [12]

TABLE I: Definition of formulation A, B, C, D', D as LIPM and E as CTM with their control inputs and origins.

We opt for formulation D and will show that the proposed state space formulation (8) has two main advantages compared with existing ones: **1.** analysis in Section IV will show that the proposed formulation is numerically more stable than the LIPM dynamics based state space formulations A, B, C, D', and thus the proposed formulation D can be used for both short and long prediction horizons; **2.** the disturbance rejection is greatly improved by introducing the new state p which will be shown in Section V.

III. Model-Predictive Control Framework

MPC enables loop closure by optimizing over a predefined prediction horizon N at a given frequency $1/T$ considering constraints. This section formulates a constrained optimization problem which is then incorporated into an MPC framework consisting of an Model-Predictive Controller, Whole-Body Controller, and a State Estimator.

A. Optimization Problem Formulation

The optimization problem, which the MPC is continuously optimizing over, is formulated as a Quadratic Programming (QP) problem due to its convex property and the existence of fast QP solvers. For this, the previously introduced continuous state space systems is discretized at a sampling time T. For any discrete Linear Time-Invariant (LTI) system

$$\boldsymbol{x}_{k+1} = A_d\boldsymbol{x}_k + B_d u_k, \quad z_k = C_d\boldsymbol{x}_k \tag{11}$$

the outputs $Z_{k+1} = [z_{k+1}, ..., z_{k+N}]^T$ at each time step can be recursively stacked up to the prediction horizon N:

$$\begin{aligned} Z_{k+1} &= \begin{bmatrix} C_d A_d^1 \\ \vdots \\ C_d A_d^N \end{bmatrix}\boldsymbol{x}_k + \begin{bmatrix} C_d A_d^0 B_d & 0 & 0 \\ C_d A_d^1 B_d & \ddots & 0 \\ C_d A_d^{N-1} B_d & \cdots & C_d A_d^0 B_d \end{bmatrix} U_k \\ Z_{k+1} &= A_t\boldsymbol{x}_k + B_t U_k. \end{aligned} \tag{12}$$

The original cost function minimizing tracking error $e_z(U) = Z(U) - Z_{Ref}$ and control input U:

$$J = (Z - Z_{ref})^T Q(Z - Z_{ref}) + U^T RU, \tag{13}$$

can be rewritten into a quadratic optimization formulation:

$$\min_U \quad \frac{1}{2}U^T HU + f^T U \tag{14}$$

$$s.t. \quad AU \leq B. \tag{15}$$

The column matrix $U \in \mathbb{R}^{N\times 1}$ contains all control actions over the prediction horizon N. The matrix $H \in \mathbb{R}^{N\times N}$ and $f \in \mathbb{R}^{N\times 1}$ are obtained by rewriting the cost function (13):

$$H = B_t^T B_t + \frac{R}{Q}\mathbb{1}_N, \quad f = B_t^T(A_t\boldsymbol{x}_k - Z_{ref}), \tag{16}$$

with identity matrix $\mathbb{1}_N$ of prediction horizon size N.

The ZMP constraints for (15) is obtained from (12):

$$Z_{min} \leq Z_{k+1}(U_k) \leq Z_{max} \tag{17}$$

$$\begin{bmatrix} -B_t \\ B_t \end{bmatrix} U \leq \begin{bmatrix} A_t\boldsymbol{x}_k - Z_{min} \\ -(A_t\boldsymbol{x}_k - Z_{max}) \end{bmatrix}. \tag{18}$$

Furthermore, for real world applications, the rate of change for the COP $\dot{p}_k = \dot{x}_k - z_c\dddot{x}_k/g$ plays a crucial role. If the optimal controller produces an unrealizable COP trajectory, the system will not be able to track it. Hence, a constraint of the rate change of the COP according to the physical system is necessary as:

$$\Delta p_{min} \leq \dot{x}_k - \frac{z_c}{g}u_k \leq \Delta p_{max} \tag{19}$$

$$\begin{bmatrix} -\frac{z_c}{g} + B_t \\ \frac{z_c}{g} - B_t \end{bmatrix} U \leq \begin{bmatrix} \Delta p_{max} - A_t\boldsymbol{x}_k \\ -(\Delta p_{min} - A_t\boldsymbol{x}_k) \end{bmatrix}, \tag{20}$$

where the output $z_k = \dot{x}_k$ is the velocity of the COM. The maximal admissible values of the COP rate change are determined by the individual robot. For Valkyrie the maximal rate change is approximately $\pm 2m/s$. Constraint (20) will be used for all CTM based QP formulations with input $u_k = \ddot{x}_k$. For LIPM based formulations, the rate change is the input $u_k = \dot{p}_k$ that can be directly constrained:

$$\Delta p_{min} \leq u_k \leq \Delta p_{max}. \tag{21}$$

B. MPC framework

The MPC framework (Fig. 3) consists of three parts: MPC, Inverse Dynamics, and State Estimator. First, the sensor information $\boldsymbol{\xi}_{sensor}$ from the sensors is passed into a state estimator [19] outputting state $\boldsymbol{x}$. Next, the Model-Predictive Controller solves the QP problem (12) for the optimal control input U using the state $\boldsymbol{x}$ obtained from the state estimator. The first element of the optimal control input $U(1) = u_k$ is used to forward simulate the dynamics (11) for the reference trajectory $\boldsymbol{\xi}_{ref}$. Lastly, an Inverse Dynamics low-level controller maps the reference COM and COP trajectories from the MPC to joint torques $\boldsymbol{\tau}$ which are then tracked on the actuators.

The torque commands $\boldsymbol{\tau}$, along with joint accelerations $\ddot{\boldsymbol{q}}$ and ground reaction forces $\boldsymbol{\lambda}$, are calculated as the solution $X = [\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}]^T$ from a whole-body QP optimization problem, presented in [7][1]:

$$\min_X X^T H X + f^T \tag{22}$$

$$s.t. \quad A_{eq} X + B_{eq} = 0 \tag{23}$$

$$A_{ineq} X + B_{ineq} \geq 0. \tag{24}$$

The cost function (22) is calculated as a weighted sum over Cartesian space acceleration tracking of COM and body links, such as swing foot and hands, COP tracking, and regularization of X. The equality constraints (23) are determined by the equations of motion:

$$\begin{bmatrix} M(q) & -S & -J^T(q) \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{bmatrix} + h(q, \dot{q}) = 0, \tag{25}$$

with inertia matrix $M(q)$, selection matrix S, stacked Jacobian matrices $J^T(q)$ of the contact links, and nonlinear effects $h(q, \dot{q})$. Torque limits, friction constraints, and COP constraints are considered in the inequality constraints (24).

IV. Analysis of Numerical Stability Related with Prediction Horizon

Guaranteeing numerical stability is an important aspect for digital control, as numerical brittleness can induce oscillations and instabilities into systems, where no noise or uncertainties exist. In optimization, if the matrices are ill-defined, nominal systems can diverge and unbounded responses can be evoked by small changes [20], [21]. Specifically, well defined matrices A_t, B_t (12) are important for the prediction

[1]For details of the implementation, please refer to [7].

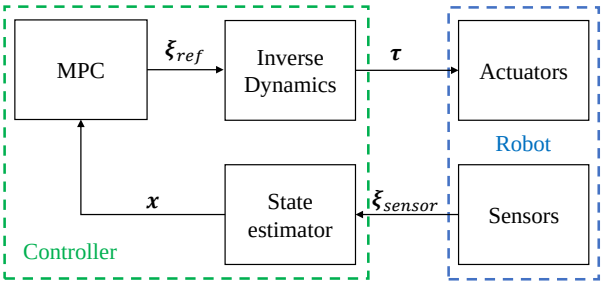


Fig. 3: Overview of the MPC framework.

matrix H in the cost function (14). Numerical issues caused by the long prediction horizon were avoided by a Singular Values Decomposition (SVD) of the optimization matrix in [20], [21]. Consequently, the constraints become softened and thus do not guarantee stability. In the following, the influence of the Prediction Horizon N on the numerical stability of different state space formulations will be analyzed, and the study will show that the numerical issues can be prevented by an appropriate choice of the dynamic model.

A. Numerical Stability

For determining the numerical stability and sensitivity the condition number $\kappa(A) = \|A\|\|A^{-1}\|$ of a matrix A will be used as an indicator. Using the Euclidian norm the calculation of the condition number becomes $\kappa(A) = \sigma_{max}(A)/\sigma_{min}(A)$, where $\sigma_{max}, \sigma_{min}$ are the maximal and minimal Singular Values (SV) respectively. As a general rule, condition numbers $\kappa(A)$ exceeding $1/\epsilon$, where ϵ is the number precision, can be considered to be an indicator for ill-defined matrices and therefore numerical brittle systems. The larger the condition number becomes, the more digit precision is lost during numerical operations, such as inverting, multiplying, etc. [22]. The condition number is exponentially influenced by the prediction horizon N. The larger N becomes, the more elements of A_d^N get exponentiated as in (12), resulting in large SV and therefore larger condition numbers. Due to this reason, it is desirable that the inter-multiplication of rows and columns in A_d are smaller than 1 to prevent the condition number to grow exponentially large. In the CTM formulation, values in the system matrix A and control matrix B are either 0 or 1. However, this is not the case for the LIPM formulation.

Fig. 4 shows the influence of the prediction horizon N on the condition number of different optimization formulations. For the matrices A_d, B_d in equation (12) the 5 formulations in Table I are considered. The matrix C_d is determined by whether COP or CP is tracked. Apart from formulation A, B, which due to its two-dimensional state space is only able to track CP alone, all other formulations are able to track both CP and COP. It can be seen that the Condition Numbers (CN) are identical for formulation A and B until the prediction horizon N exceeds 5s,where the CN exceeds $1/\epsilon$ and thus fluctuates due to numerical inaccuracies.

This phenomenon can be observed in all LIPM based state space representations: the function of CN over prediction

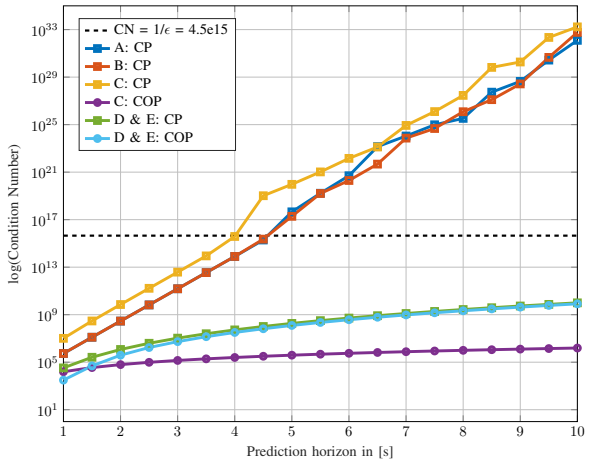


Fig. 4: Logarithmic representation of the condition number (CN) over different prediction horizons. Squared points are CN for CP tracking, circled points are CN for COP tracking. The CN for formulation E and proposed D overlap in both CP and COP.

Formulation		Prediction Horizon NT 1.5s	2.5s	3.5s	4.5s	5.5s
A: CP	CN	1e07	7e09	3e12	2e15	**2e19**
	EV	0.97	0.97	0.97	0.97	**68.74**
B: CP	CN	1e07	7e09	3e12	2e15	**2e19**
	EV	1.03	1.03	1.03	1.03	**8.81**
C: CP	CN	3e08	2e11	9e13	**1e19**	**1e21**
	EV	0.97	0.97	0.97	**1873.3**	**216.3**
C: COP	CN	4e04	1e05	2e05	3e05	5e05
	EV	**1.03**	**1.03**	**1.03**	**1.03**	**1.03**
D: CP	CN	3e05	4e06	2e07	1e08	3e08
	EV	1.00	1.00	1.00	1.00	1.00
D: COP	CN	5e04	2e06	1e07	7e07	2e08
	EV	1.00	1.00	1.00	1.00	1.00
E: CP	CN	3e05	4e06	2e07	1e08	3e08
	EV	0.97	0.97	0.97	0.97	0.97
E: COP	CN	5e04	2e06	1e07	7e07	2e08
	EV	0.97	0.97	0.97	0.97	0.97

TABLE II: Condition Number (CN) and Eigenvalue (EV) over different prediction horizons NT for the formulations presented in Section II, and COP or CP tracking.

horizon follows an exponential curve, but after the CN exceeds $1/\epsilon$, the CN fluctuates, while still increasing exponentially. An exception is the COP tracking formulation C: the matrices are well-conditioned for output matrix $C_d = [0, 0, 1]$ and state vector $\boldsymbol{x} = [x, \dot{x}, p]^T$. The CTM formulation E, and the proposed formulation have well-defined matrices, and the CN only exceeds $1/\epsilon$ after more than a 100s.

B. Open Loop Stability for Planning

To further evaluate whether the state space formulations are suitable for planning, i.e. optimization over a long prediction horizon, the Eigenvalues (EV) were analyzed. As in Table II, the prediction horizon does not influence the EV of the CTM based representations. For LIPM based representations however, the prediction horizon influences the EV after $NT = 5.5$s making these models unsuitable for long term planning as the open-loop case is highly unstable.

In Fig. 5, the effect of a prediction horizon $T_{prev} = 5.5s$ for solution U of the planning problem (14) is shown exemplary (all LIPM formulations exhibit this behavior) for CP tracking using formulation C and our proposed formulation D. The yellow line is the COP rate change trajectory that achieves perfect tracking given the CP reference.

It can be seen that the optimal solution using the LIPM (blue line) performs poorly, due to oscillation and deviations from the reference. This is caused by large prediction horizons, which can be observed by analyzing the last row of the prediction matrix B_t in (12). It describes the contribution of each input $u_k, k = [1, ..., N]$ to the final output z_N. As shown in Fig. 5, the control inputs of LIPM based formulation have almost no effect on the final state after $2s$. In contrast, our proposed formulation and formulation A (red line) are unaffected by the prediction horizon. All inputs u_k influence the final state with decreasing importance as k increases.

The comparative analysis shows that LIPM based models are not suitable for long term planning. If the loop is closed by the optimal solution of such models with large prediction

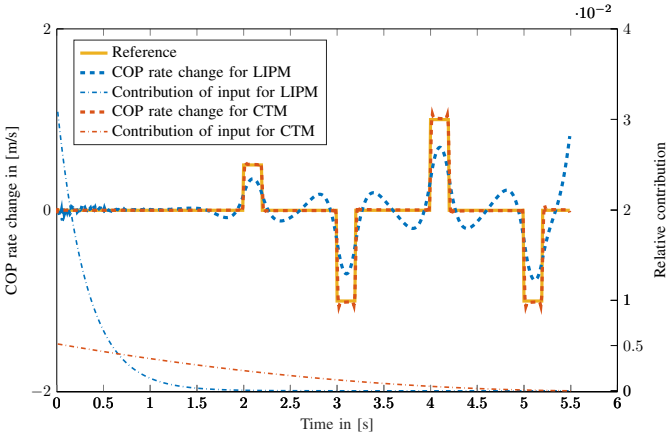


Fig. 5: COP rate change trajectories (left y-axis) for LIPM and CTM. The relative contribution (right y-axis) of input u_k to the final output z_N is in dashed lines for LIPM (blue) and CTM (red).

horizons (empirically maximum tested: $4s$), the computed control input (COP) oscillates and ultimately diverges the system. This is caused, beside oscillations, by the ill-defined prediction matrices B_t result in inaccurately calculated feedback gains K, and therefore destabilize the system.

Summarized, all CTM based optimizations are suitable for long prediction horizons and therefore gait planning. The gait planner generates a COM trajectory that follows the ZMP or the CP reference. Due to the well-defined prediction matrices A_t, B_t, the proposed formulation D and E can generate walking patterns to up to 100s. Given a ZMP or CP reference, an one-shot solution of the optimization can then be used as a nominal gait planning. The limitation of existing CTM based MPC schemes will be studied in the next section.

V. Simulation Benchmarks

This section presents the comparison study of closed loop behavior for different formulations C, E, and our proposed formulation D in COP and CP tracking tasks while applying a variety of impulse and constant disturbances.

A. Simulation setup

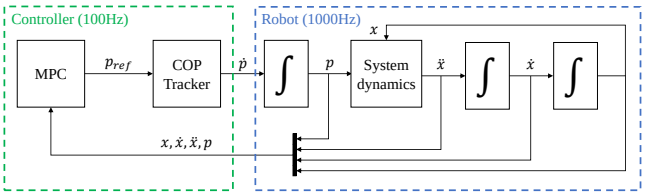


Fig. 6: Overview of the simulation setup in MATLAB.

For simulating the physics, two different platforms are chosen: Gazebo and MATLAB. The validity of the proposed MPC framework (Section III) for gait planning and feedback control is shown in the real world simulator Gazebo (Fig. 1). The framework is able to handle unmodelled inclined terrain of up to $5°$ pitch and $10°$ roll inclination. The results can be seen on `https://youtu.be/7uwtorW_kzE`.

For benchmarking the disturbance rejection abilities a numerical simulation is conducted in MATLAB. The robot's dynamics (Fig. 6) are numerically integrated at $T_{sim} = 1$ms, and the closed-loop MPC is running at a frequency of $f_{mpc} = 100Hz$. The COM dynamics follow the "COP→ acceleration" (1) causality determined by the LIPM (Fig. 2):

$$\ddot{x}_{com} = \frac{g}{z_c}(x_{com} - p^*) + \ddot{x}_{ext}. \tag{26}$$

Formulation C expresses identical dynamics as formulation A and B, and can therefore be seen as a representative extension of the latter. The objective (13) of the optimization is tracking a predetermined ZMP or CP trajectory under constraints. For numerically sensitive systems, correct selection of parameters is crucial: short prediction horizons or over-constrained inputs ($R < 10e^{-6}$) may result in instability. The prediction horizon was chosen as $N = 2.5s$, the weights as $Q = 1, R = 10e^{-6}$. The system is considered to be unstable, if the states diverge, the ZMP exceeds the SP (e.g. Fig. 8a), or no solution of the optimization problem is found, i.e. the COP rate change constraint is significantly violated.

B. Disturbance Rejection

To determine the capability of disturbance rejection for the different formulations with different tracking objectives, four types of disturbances are studied: three impulses at different impact times, and one constant disturbance. All types of disturbances have relevance in the real application: impulse disturbances can be considered as pushes, or sudden impacts caused by collision, e.g. improper swing foot landing. Constant acceleration disturbances can be caused by an external load, biased COM position from the state estimation errors [19], or model uncertainties. The maximal rejectable disturbance of the individual models is detailed in Table III.

Alternatively for formulation E, instead of applying the erroneously calculated ZMP, it is possible to simply apply clipping of the output ZMP/COP when it exceeds the SP as the work in [23]. Even though clipping can prevent the system from getting unstable due to the ZMP exceeding the SP, the system can still become unstable due to divergent motion, or insufficient COP rate change. This approach is still inferior to our proposed formulation in theory, because the constraints are not correctly represented in the formulation and hence the clipped control effort is inevitably suboptimal, which leads to roughly 50% less disturbance rejection ability compared to that of our proposed formulation.

	Disturbance duration in seconds			
Formulation	0–10	2.4–2.45	2.6–2.65	2.8–2.85
C (CP)	0.36 (0.31)	2.04 (0.91)	2.57 (0.92)	2.02 (0.80)
C (COP)	0.18 (0.15)	**2.23 (1.00)**	**2.78 (1.00)**	2.23 (0.89)
D (CP)	0.50 (0.43)	2.04 (0.91)	2.57 (0.92)	2.29 (0.91)
D (COP)	**1.17 (1.00)**	2.16 (0.97)	2.76 (0.99)	**2.51 (1.00)**
E (CP)	0.42 (0.36)	0.30 (0.13)	0.37 (0.13)	0.34 (0.14)
E (COP)	0.56 (0.48)	0.35 (0.16)	0.42 (0.15)	0.44 (0.18)
E clip. (CP)	0.35 (0.30)	1.42 (0.64)	1.40 (0.50)	1.40 (0.56)
E clip. (COP)	0.56 (0.48)	1.39 (0.62)	1.39 (0.50)	1.37 (0.55)

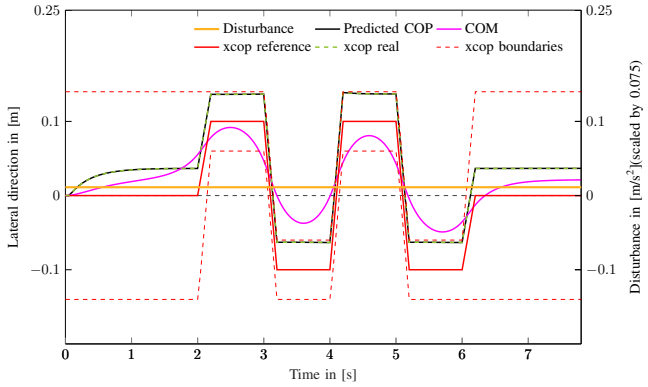
TABLE III: Maximal rejectable disturbance for CP or COP tracking. The ratio in the brackets is normalized by the maximal disturbance (bold) for the specific disturbance type and formulation.

C. Performance Analysis

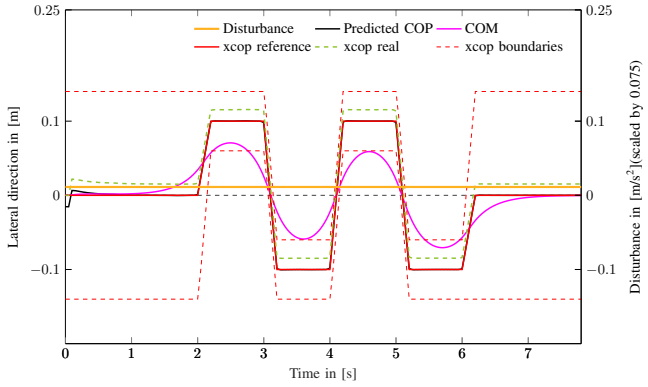
1) Comparison of Models: Under constant disturbance our proposed formulation performs twice as good as the others (Table III). Our proposed MPC withstands more than double the amount of disturbance compared to the other formulations in COP tracking tasks, and withstand averagely 34% more perturbation in CP tracking tasks. Compared to other formulations, the disturbance is handled better, and therefore the COP rate change only hits the constraints at much larger perturbations. Formulation C fails due to the violation of the COP rate change constraints, and the ZMP in formulation E violates the SP constraints earlier (Fig.7).

For constant disturbances, the proposed formulation D is able to distinguish between disturbance-generated and self-generated accelerations and therefore correctly calculating the COP that respects the physical constraints. By doing so, it is able to counterbalance the constant disturbance, track the COP reference with zero steady state error after 1 second by offsetting the COM away from the nominal trajectory and leaning against the constant push (cf. behavior after 7s). In contrast, formulation E generates the COP away from the desired reference with steady state error while keeping the COM at the nominal trajectory yielding.

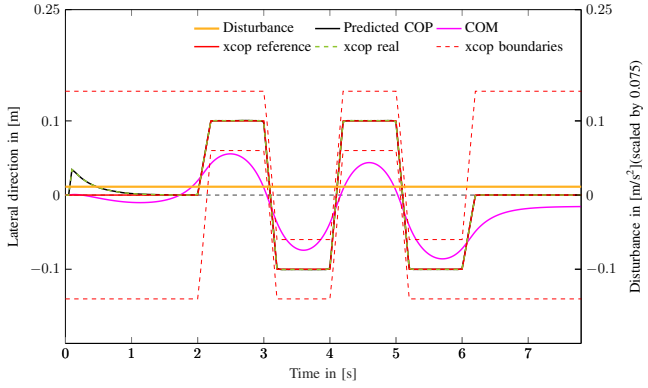
Formulation C shifts both COP and COM away from the nominal trajectory, because only the input $\dot{p}$ is used to track the COP reference without considering the COM state. However, convergence is only guaranteed by including the COM states in the cost function or constraints. For short disturbances, e.g. impulses, deviations between predicted COM state and real disturbed COM state do not influence the performance of the controller. But for long, constant deviations, ignoring the COM state leads to decreased performance. This can be seen in Fig. 7(b). It can be further elaborated by turning the constant disturbance off, and observing how the steady state error vanishes, as predicted COM state and real COM state coincide again (cf. attached video).



(a) Constant disturbance while COP tracking with formulation C.



(b) Constant disturbance while COP tracking with formulation E.

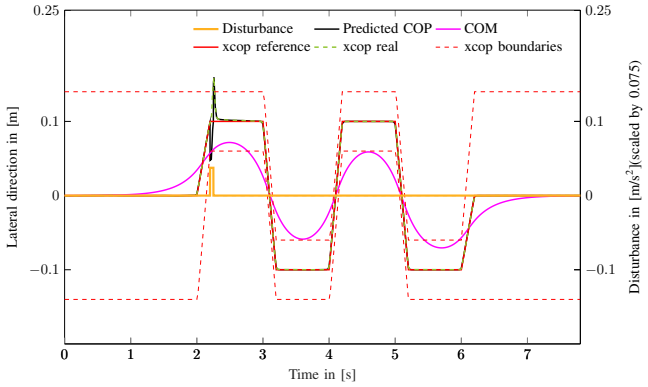


(c) Constant disturbance while COP tracking with proposed formulation D.

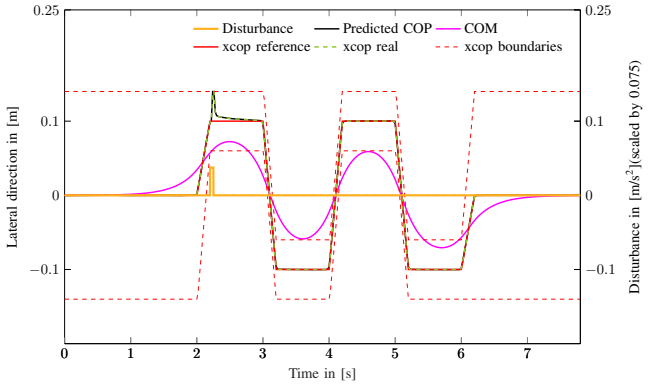
Fig. 7: Comparison between formulation C (a), E (b), and proposed D (c) for constant disturbance $\ddot{x}_{ext} = 0.15m/s^2$. The proposed formulation D updates the COP trajectory correctly according to (8). Formulation E calculates the COP trajectory wrongly by (2), attributing the external acceleration wrongly to the COP.

For impulse disturbances (Fig. 8), the time of impact is almost irrelevant with respect to the ratio (see Table III). Although satisfying the internal model of the ZMP constraint, formulation E becomes unstable because of the violation of the SP constraint due to an erroneously calculated ZMP (cf. Section II-B). Both the proposed formulation D and C can handle the ZMP dynamics correctly with almost the same performances that are better than formulation E.

In summary, the disturbance rejection capability is not fully maximized in the CTM based MPC formulation E. LIPM based MPC formulations perform similar to the proposed formulation for impulse disturbances, but reject less performance during constant disturbances.



(a) Impulse disturbance while COP tracking with formulation E.



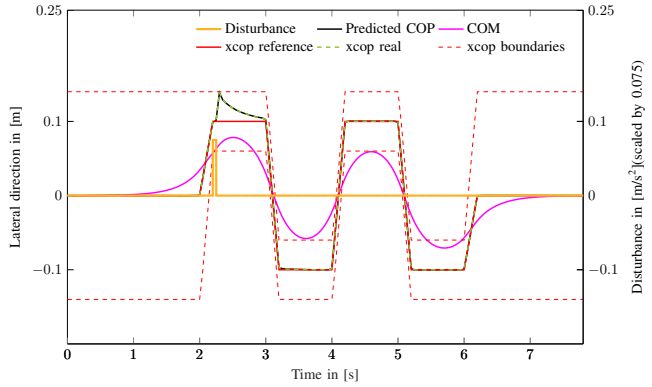
(b) Impulse disturbance while COP tracking with proposed formulation D.

Fig. 8: Comparison between formulation E (a), and proposed D (b) for an impulse push of $\ddot{x}_{ext} = 0.5m/s^2$. Due to wrongly capturing the COP dynamics the controller in (a) is wrongly reacting to the disturbance, moving the COP outside of the SP. Formulation (b) is correctly complying with the ZMP constraint.

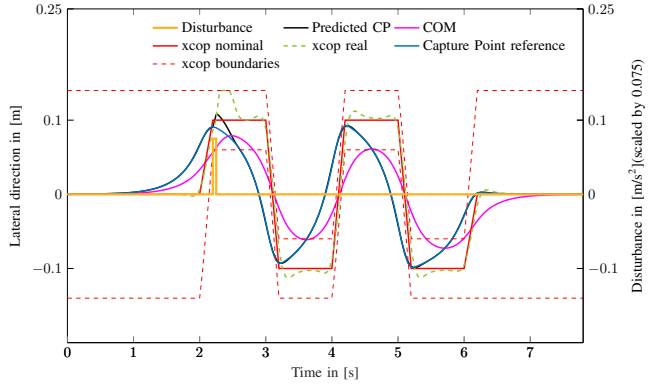
2) Comparison of Control Objectives: Although our former study shows that the COP tracking has good performance against disturbances, decomposing the COM's motion into a stable and unstable parts and controlling the unstable part of divergent component or capture point is known for additional advantages [24].

Setting the control objective of minimizing CP, i.e. CP tracking, makes the system fully utilizing the stability margin while being disturbed by quickly moving the COP to the boundaries as possible. In contrast, an objective of minimizing COP error, i.e., COP tracking, limits the deviation of COP away from the reference, thus a slower COP response in the presence of disturbance, so a problem may arise if a second disturbance occurs.

The CP tracking recovers faster and can therefore possibly handle a new impulse, whereas the COP tracking recovers slower being vulnerable to a new push. The response of an impulsive push using the proposed formulation can be seen Fig. 9. After getting disturbed for 50ms at $t = 2.20$s, the CP tracking has COP settles back at $t = 2.5$s, while for the COP tracking the COP is still at 30% of its stability margin.



(a) COP tracking with proposed formulation D.



(b) CP tracking with proposed formulation D.

Fig. 9: Comparison between COP (a) and CP (b) tracking under an impulse disturbance of $\ddot{x}_{ext} = 1m/s^2$.

Besides, the COP tracking scheme reinforces smaller COP errors due to its objective and thus demands larger COP rate change; while the CP scheme allows small harmless variations of COP as long as CP is tracked and COP is feasible, thus requires smaller COP rate change.

VI. Conclusion

This paper has proposed a formulation for Model-Predictive Control based on the Linear Inverted Pendulum, which has an improved performance over previous formulations and is suitable for both gait planning and feedback control of legged locomotion. It resolves two previously unattended problems: (1) numerical stability in the case of long prediction horizons, making the proposed method more suitable for gait planning; (2) enhanced disturbance rejection attributed to the incorporation of COP as a new state variable, and therefore correctly including the COP in the feedback loop. Our study showed that the proposed formulation was able to reject larger disturbances than the previous LIP formulations. Furthermore, in combination with a whole-body controller, the proposed MPC framework was able to blindly traverse unknown inclined terrain of up to 5° and 10° for pitch and roll inclination respectively.

Acknowledgement

This work is supported by the UK Robotics and Artificial Intelligence Hubs – Future AI and Robotics for Space (FAIR-SPACE), EP/R026092/1 and Offshore Robotics for Certification of Assets (ORCA), EP/R026173/1 – funded by the Engineering and Physical Sciences Research Council (EPSRC), UK.

References

[1] C. G. Atkeson, B. P. Babu *et al.*, "What Happened at the DARPA Robotics Challenge and Why," 2015.

[2] M. Johnson, B. Shrewsbury *et al.*, "Team IHMC's Lessons Learned from the DARPA Robotics Challenge: Finding Data in the Rubble," *Journal of Field Robotics*, vol. 34, pp. 1–17, 2015.

[3] S. Kuindersma, R. Deits *et al.*, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, pp. 429–455, 2016.

[4] D. Q. Mayne, J. B. Rawlings *et al.*, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, 2000.

[5] M. Vukobratović and B. Borovac, "Zero-Moment Point Thirty Five Years of Its Life," *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, 2004.

[6] S. Kajita and K. Tani, "Study of dynamic biped locomotion on rugged terrain-derivation and application of the linear inverted pendulum mode," *International Conference on Robotics and Automation*, 1991.

[7] S. Feng, X. Xinjilefu *et al.*, "3D Walking Based on Online Optimization," *International Conference on Humanoid Robots*, 2013.

[8] R. Tedrake, S. Kuindersma *et al.*, "A closed-form solution for real-time optimal gait generation and feedback stabilization," *International Conference on Humanoid Robots (Humanoids)*, 2015.

[9] A. W. Winkler, F. Farshidian *et al.*, "Online walking motion and foothold optimization for quadruped locomotion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[10] M. Morisawa, S. Kajita *et al.*, "Balance control based on Capture Point error compensation for biped walking on uneven terrain," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012.

[11] S. Kajita, F. Kanehiro *et al.*, "Biped walking pattern generation by using preview control of zero-moment point," *IEEE International Conference on Robotics and Automation (ICRA)*, 2003.

[12] P. B. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2006.

[13] A. Herdt, H. Diedam *et al.*, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, 2010.

[14] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010.

[15] M. Krause, J. Englsberger *et al.*, "Stabilization of the Capture Point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, pp. 165–171, 2012.

[16] R. J. Griffin and A. Leonessa, "Model predictive control for dynamic footstep adjustment using the divergent component of motion," *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[17] J. Englsberger, C. Ott *et al.*, "Bipedal walking control based on capture point dynamics," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[18] S. Kajita, M. Morisawa *et al.*, "Biped Walking Pattern Generator allowing Auxiliary ZMP Control," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[19] B. J. Stephens, "State Estimation for Force-Controlled Humanoid Balance using Simple Models in the Presence of Modeling Error," *IEEE International Conference on Robotics and Automation*, 2011.

[20] J. Castano, A. Hernandez *et al.*, "Implementation of robust EPSAC on dynamic walking of COMAN humanoid," *International Federation of Automatic Control (IFAC)*, vol. 47, pp. 8384–8390, 2014.

[21] J. A. Castano, A. Hernandez *et al.*, "Enhancing the robustness of the EPSAC predictive control using a Singular Value Decomposition approach," *Robotics and Autonomous Systems*, vol. 74, 2015.

[22] E. W. Cheney and D. R. Kincaid, *Numerical Mathematics and Computing*. Brooks/Cole Publishing Co., 2007.

[23] J. A. Castano, Z. Li *et al.*, "Dynamic and Reactive Walking for Humanoid Robots Based on Foot Placement Control," *International Journal of Humanoid Robotics*, vol. 13, 2016.

[24] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion," *IEEE Transactions on Robotics*, vol. 31, pp. 355–368, 2015.